



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/064,468	07/17/2002	Ying-Chou Chen	ASIP0002USA	5821

27765 7590 04/08/2005

NORTH AMERICA INTERNATIONAL PATENT OFFICE (NAIPC)  
P.O. BOX 506  
MERRIFIELD, VA 22116

EXAMINER

STEELMAN, MARY J

ART UNIT PAPER NUMBER

2191

DATE MAILED: 04/08/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>		<b>Applicant(s)</b>	
	10/064,468		CHEN, YING-CHOU	
	<b>Examiner</b>		<b>Art Unit</b>	
	Mary J. Steelman		2191	

**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 17 July 2002.
- 2a) ☐ This action is FINAL.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-19 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-19 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 17 July 2002 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)  | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                                   | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)             |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

### **DETAILED ACTION**

1. Claims 1-19 are pending.

#### ***Claim Objections***

2. 7.35.01 Trademark or Trade Name as a Limitation in the Claim

Claims 8, 10, and 19 contains the trademark/trade name Linux. Where a trademark or trade name is used in a claim as a limitation to identify or describe a particular material or product, the claim does not comply with the requirements of 35 U.S.C. 112, second paragraph. See Ex parte Simpson, 218 USPQ 1020 (Bd. App. 1982). The claim scope is uncertain since the trademark or trade name cannot be used properly to identify any particular material or product. A trademark or trade name is used to identify a source of goods, and not the goods themselves. Thus, a trademark or trade name does not identify or describe the goods associated with the trademark or trade name. In the present case, the trademark/trade name is used to identify/describe an operating system and, accordingly, the identification/description is indefinite.

#### ***Specification***

3. The use of the trademark Linux has been noted in this application. It should be capitalized wherever it appears and be accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner which might adversely affect their validity as trademarks.

***Claim Rejections - 35 USC § 103***

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 1-3, 7, 9, 11-14, and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Pre Grant Pub 2002012078 A1 to Plaxton et al.

Per claim 1:

A program system stored in a memory of a computer system comprising: at least one computer program:

Plaxton: [0029], "a device and additional program files (program system stored in a memory of a computer system) having an arbitrary number of n class files...user desires to add to device..."

-and a software library having at least one first-type subroutine module and at least one second-type subroutine module, where the first-type subroutine module is used by the computer program and the second-type subroutine module is not used by the computer program;

Plaxton: Plaxton disclosed a "first-type subroutine module", which processes code to be used by the system and a "second-type subroutine module", which processes code not used by the system. See FIG. 3. Preloaded Application and libraries (stored in permanent memory) may be pre-internalized into a platform specific image and stored in a section of permanent memory (a first-type subroutine). [0036], A Java Application Manager (JAM) may direct a second or third

Art Unit: 2191

set of updates to be loaded, pre-internalized and stored in permanent memory. "...the JAM directs the Virtual machine to remove the second set of program files from the permanent memory (JAM removes unused files from the permanent memory / second-type subroutine module not used by the computer program). [0031], "The application manager functions to assist with placement of the additional files in either dynamic memory or permanent memory ...pre-internalized image of the program files...stored in permanent memory...the original copy of the additional files placed into dynamic memory or permanent memory can now be deleted (not used by the computer program)...(application manager can also remove original copy of files, saving only the pre-internalized image)"

-wherein after the software library has passed a compilation process first and then a linkage process, the processed software library is stored in the memory of the computer system, and the second-type subroutine module in the processed software library is changed in a non-recoverable manner after the linkage process so that the memory required to store the second-type subroutine module is saved and can be used by the computer system,

Plaxton: [0031] "pre-internalize the program files into the Virtual Machine's native memory structure (compile), [0032], "The class loader (link and load) and class resolver have been modified to execute in either normal mode or pre-internalization mode", [0031], "pre-internalized image...stored in permanent memory (stored in memory of the computer system", [0031], "original copy of the additional files placed into dynamic memory or permanent memory can now be deleted (changed in a non-recoverable manner)..." Plaxton stores the pre-internalized image (see [0009]: "Pre-internalization is a process of reformatting Java class file

Art Unit: 2191

information into a format that when placed in memory represents a class object...reducing dynamic memory storage. The format of the pre-internalized file is specific to each Java VM implementation.”) in a permanent memory and may delete the original, imported upgraded files.

-and when the computer program is updated later on to use the second-type subroutine module, the second-type subroutine module is stored to the processed software library so that the updated computer program can use both the first-type and second-type subroutine modules in the software library.

Plaxton: When an update to the program requires use of the original unused files, they may be found (in a non-pre-internalized state) in a permanent memory location. See FIG. 5, an original file that had been pre-internalized can be removed (#107), however, it remains in its “preloaded application and libraries”, #68, FIG.3, and can again be made into a “pre-internalized” image in permanent memory, (#84, FIG.4) An updated program can use original files stored in permanent memory by using the pre-internalizer to create an image [0036], or can use a pre-internalized image that was previously stored in permanent memory.

Plaxton disclosed that ‘preloaded application and libraries may be stored in permanent memory (FIG. 3), then proceeded to explain that ‘pre-internalized’ images may be stored in permanent memory. Additional ‘pre-internalized’ images may be added for an update, or ‘pre-internalized’ images may be removed as needed to manage memory. The preloaded files may again be transformed and stored as a ‘pre-internalized’ image in permanent memory, as needed, by the Java Application manager. Plaxton recognized the need to update embedded systems and the

Art Unit: 2191

concerns for limited memory. Although Plaxton did not use the same words as the claimed limitation, his invention broadly disclosed Applicant's claimed limitation.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention to modify Plaxton's invention to more distinctly point out that a first-type of subroutine module broadly includes pre-internalized images of code, permanently stored, that will be used by a system, and a second-type of subroutine module broadly includes code that is removed / or not pre-internalized (not permanently stored in a pre-internalized format) because it is not used. Therefore memory is better managed, which is important for small embedded systems.

Per claim 2:

-when storing the second-type subroutine module to the software library when updating the computer program, the first-type subroutine module can be changed in a non-recoverable manner and the second-type subroutine module stored to the software library can be used by the updated computer program directly without undergoing another compilation and linkage process first.

Plaxton: Application Manager can remove [0038] the 'first-type subroutine module' (a pre-internalized image changed in a non recoverable manner), and can update and store a 'second-type subroutine module' by downloading and pre-internalizing files [0036], thereby storing into permanent memory. [0036], "JAM (Java Application Manager), directs Virtual Machine to pre-internalize the third set of program files (update files) by primarily using its pre-internalizer...create an image...that is then stored..."

Per claim 3:

-a plurality of computer programs and the software library comprising a plurality of first-type subroutine modules wherein each of the first-type subroutine modules is used by at least one computer program and the second-type subroutine module is not used by any of the computer programs.

Plaxton: [0038], "FIG. 5 is a process illustrating a variety of program application execution options for a Java application manager..." From a plurality of computer programs (#103), a pre-internalized image can be created (#124), and used (first-type, #116, #109) or removed (second-type, #112).

Per claim 7:

-the computer system comprises an operating system for controlling operations of the computer system,

Plaxton: [0030], "virtual machine (VM)...( operating system for controlling operations of the computer system)" The virtual machine acts as an operating system.

-and the first-type and second-type subroutine modules are stored in the memory of the computer system through the operating system,

Plaxton: See FIG. 4. First-type subroutine modules (code that is used by system) is stored in a pre-internalized format. Second-type subroutine modules (code that is not used by system) is stored in #81, preloaded classes.



Art Unit: 2191

-and wherein the memory used by the second-type subroutine module can be released by the operating system for other purposes and the second-type subroutine module is non-recoverable after the release.

Plaxton: [0046], "...either moving or totally removing (is non-recoverable) a pre-internalized program file from memory."

Per claim 9:

-the computer system comprises an operating system for controlling operations of the computer system,

Plaxton: [0030], "virtual machine (VM)" A virtual machine is used for controlling operations of the computer system,.

-and the first-type subroutine module is stored in the memory of the computer system through the operating system after the linkage process while the second-type subroutine module is not stored so as to save memory space.

(See rejection of limitations in claim 1 above.) Pre-internalized images are stored, while unneeded pre-internalized images may be removed [0038] to save memory space.

Per claim 11:

-the computer system is an embedded computer system and the memory is formed by using only memory IC (integrated circuit) chips.

Plaxton: [0030], "permanent memory may be implemented as a Flash memory (IC chip)..."

Per claim 12:

A program system in an embedded system comprising: a computer program stored in a memory of the embedded system;

Plaxton: [0029], "device and additional program files...", [0027], "embedded..."

-and a software library having a first-type subroutine module and a second-type subroutine module, where the first-type subroutine module is used by the computer program and the second-type subroutine module is not used by the computer program;

-wherein the software library after a compilation process is stored in the memory, and the second-type subroutine module in the processed software library is changed in a non-capacity manner after a linkage process so that the memory required to store the second-type subroutine module is saved and can be used by the embedded system, and when the computer program is updated later on to use the second-type subroutine module, the second-type subroutine module is stored to the processed software library so that the updated computer program can use both the first-type and second-type subroutine modules in the software library.

This is an "embedded system" version of claim 1. See rejection of claim 1 above.

Per claim 13:

-when storing the second-type subroutine module to the software library when updating the

Art Unit: 2191

computer program, the first-type subroutine module can be changed in a non-capacity manner and the second-type subroutine module stored to the software library can be used by the updated computer program directly without undergoing another compilation and linkage process first.

See rejection of limitations as addressed in claim 2 above.

Per claim 14:

-a plurality of computer programs and the software library comprising a plurality of first-type subroutine modules wherein each of the first-type subroutine modules is used by at least one computer program and the second-type subroutine module is not used by any of the computer programs.

See rejection of limitations as addressed in claim 3 above.

Per claim 18:

-the embedded system comprises an operating system for controlling operations of the embedded system, and the first-type and second-type subroutine modules are stored in the memory of the embedded system through the operating system, and wherein the memory used by the second-type subroutine module can be released by the operating system for other purposes and the second-type subroutine module is non-recoverable after the release.

See rejection of limitations as addressed in claim 9 above.

6. Claims 4-6 and 15-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over US PreGrant Pub 20020129078 A1 to Plaxton et al., in view of US Patent 6,832,373 B2 to O'Neill.

Per claims 4 and 5:

- the second-type subroutine module is changed to an easy-compression format in a non-recoverable manner after the linkage process, and the changed second-type subroutine module is compressed first before storing to the memory of the computer system to save memory space.

- the second-type subroutine module is changed to an easy-compression format in a non-recoverable manner after the linkage process, and the software library is compressed first before storing to the memory of the computer system to save memory space.

Plaxton disclosed saving program files in first-type and second-type modules, and changing a format in a non-recoverable manner (removing a pre-internalized format). Plaxton failed to disclose storing modules (including libraries) in a compressed format. However, O'Neill disclosed (Abstract, lines 1-7) updating and storing digital information to embedded storage locations. O'Neill disclosed (col. 15, line 61-col. 16, line 14), "Alternatively, in the state 262, if the client device 104 determines that there is not enough memory or storage space available or allocated in the client device 104 to accommodate the update package 110, then the client device 104 submits a request, in a state 270 to perform an allocation procedure where additional memory or storage space is freed up to accommodate the download transfer of the update package 110. In one embodiment, to allocate space for the download transfer of the update package 104, the client device 104 may write current files stored in a first data area (i.e. RAM) to a second data area (i.e. onboard flash memory). Alternatively, the client device 104 may compress the files stored (an easy compression format) in the first data area to make more space

Art Unit: 2191

available for the update package. To further allocate memory, the client device 104 may transfer files ... It will be appreciated that a combination of the aforementioned memory allocation schemes may be used to create sufficient room to receive the update package 110. Additionally, other memory allocation schemes may be used without departing from the scope of the present invention.

Therefore, it would have been obvious to one of ordinary skill in the art, at the time of the invention to have modified Plaxton's invention by including various compression and memory allocation schemes, as disclosed by O'Neill, because Plaxton recognized the [0004] memory constraint for embedded devices and the need to update applications to [0015] add supplemental functionality, making embedded devices more useful to consumers.

Per claim 6:

-the software library is compressed first and the computer system comprises an operating system for controlling operations of the computer system, and when the operating system executes the compressed first-type subroutine module, it will de-compress the first-type subroutine module first.

Plaxton disclosed saving program files in first-type and second-type modules, and changing a format in a non-recoverable manner (removing a pre-internalized format). Plaxton failed to disclose de-compressing memory as needed for execution. However, O'Neill disclosed (col. 28, lines 39-45), "The update agent 1025 may also include functionality for performing various

Art Unit: 2191

operations used to compress data in the data section to create sufficient storage space in the data section to receive the update package 110. Finally, the update agent 1025 may include functionality used to prepare the update package 110 such as compression/decompression...”

Therefore, it would have been obvious to one of ordinary skill in the art, at the time of the invention to have modified Plaxton's invention by including various compression / decompression and memory allocation schemes, as disclosed by O'Neill, because Plaxton recognized the [0004] memory constraint for embedded devices and the need to update applications to [0015] add supplemental functionality, making embedded devices more useful to consumers.

Per claim 15:

-the second-type subroutine module is changed to an easy-compression format in a non-capacity manner after the linkage process, and the software library is compressed first before storing to the memory of the embedded system to save memory space.

See rejection of limitations as addressed in claim 4 above.

Per claim 16:

-the second-type subroutine module is changed to an easy-compression format in a non-capacity manner after the linkage process, and the software library is compressed first before storing to the memory of the embedded system to save memory space.

See rejection of limitations as addressed in claim 5 above.

Per claim 17:

-the software library is compressed first and the embedded system comprises an operating system for controlling operations of the computer system, and when the operating system executes the compressed first-type subroutine module, it will de-compress the first-type subroutine module first.

See rejection of limitations as addressed in claim 6 above.

7. Claims 8, 10, and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over US PreGrant Pub 20020129078 A1 to Plaxton et al., in view of US Patent 5,867,712 A to Shaw.

Per claims 8, 10 and 19:

-the operating system is a Linux system.

Plaxton failed to suggest a Linux system. However, Shaw suggested that a UNIX (Linux) application may be embedded. Shaw disclosed (col. 1, lines 39-53), "These CISC or RISC host processing or coprocessing techniques can partially improve the performance of specific data subsystems, such as encoding multiple algorithms, managing memory or display devices, and adapting to existing DOS, OS2, WINDOW, NT, or UNIX (Linux) application and system environments. Typically, they can be readily implemented either in hardware, firmware, or software means embedded with custom integrated circuit, digital signal processor, or application specific integrated circuit (ASIC's). Shaw disclosed (col. 4, lines 31-38), "In FIG. 3, DISC architecture also illustrate an embedded RISC or CISC co-processor element in order to

Art Unit: 2191

directly execute the bit oriented application programs in DOS, Window, NT, Macintosh, OS2, UNIX, or alike. In a more preferred embodiment, DISC can include a real time object oriented operation system wherein concurrent execution of the application program and real time DISC based document computing can be performed.

Therefore, it would have been obvious to one of ordinary skill in the art, at the time of the invention to have modified Plaxton's invention by disclosing that an embedded application could also exist on a chip able to operate a Linux system, as suggested by Shaw, because Plaxton disclosed [0002], "This invention relates generally to processing of program files written in a high-level language..." and Linux is a high level language. Regardless of what language the system is written in, an embedded system may be updated to make the device more useful.

### *Conclusion*

8. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (571) 272-3704. The examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached at (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.



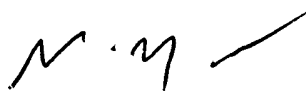
Art Unit: 2191

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mary Steelman



03/31/2005



WEI Y. ZHEN  
PRIMARY EXAMINER